Clusterpath An Algorithm for Clustering using Convex Fusion Penalties

Toby Dylan Hocking, Armand Joulin, Francis Bach, Jean-Philippe Vert

▶ **To cite this version:**

Toby Dylan Hocking, Armand Joulin, Francis Bach, Jean-Philippe Vert. Clusterpath An Algorithm for Clustering using Convex Fusion Penalties. 28th international conference on machine learning, Jun 2011, United States. pp.1. hal-00591630

HAL Id: hal-00591630
https://hal.science/hal-00591630

Submitted on 9 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Clusterpath
An Algorithm for Clustering using Convex Fusion Penalties

Toby Dylan Hocking – Toby.Hocking@inria.fr
INRIA – Sierra team, Ecole Normale Supérieure, Mines ParisTech – CBIO, INSERM U900, Institut Curie, Paris, France

Armand Joulin – Armand.Joulin@inria.fr
INRIA – Sierra team, Ecole Normale Supérieure, Paris, France

Francis Bach – Francis.Bach@inria.fr
INRIA – Sierra team, Ecole Normale Supérieure, Paris, France

Jean-Philippe Vert – Jean-Philippe.Vert@mines.org
Mines ParisTech – CBIO, INSERM U900, Institut Curie, Paris, France

May 9, 2011

Abstract

We present a new clustering algorithm by proposing a convex relaxation of hierarchical clustering, which results in a family of objective functions with a natural geometric interpretation. We give efficient algorithms for calculating the continuous regularization path of solutions, and discuss relative advantages of the parameters. Our method experimentally gives state-of-the-art results similar to spectral clustering for non-convex clusters, and has the added benefit of learning a tree structure from the data.

Contents

1 Introduction

In the analysis of multivariate data, cluster analysis is a family of unsupervised learning techniques that allows identification of homogenous subsets of data. Algorithms such as k-means, Gaussian mixture models, hierarchical clustering, and spectral clustering allow recognition of a variety of cluster shapes. However, all of these methods suffer from instabilities, either because they are cast as non-convex optimization problems, or because they rely on hard thresholding of distances. Several convex clustering methods have been proposed, but some only focus on the 2-class problem [XNLS04], and others require arbitrary fixing of minimal cluster sizes in advance [BH08]. The main contribution of this work is the development of a new convex hierarchical clustering algorithm that attempts to address these concerns.

In recent years, sparsity-inducing norms have emerged as flexible tools that allow variable selection in penalized linear models. The Lasso and group Lasso are now well-known models that enforce sparsity or group-wise sparsity in the estimated coefficients [Tib96, YL06]. Another example, more useful for clustering, is the fused Lasso signal approximator (FLSA), which has been used for segmentation and image denoising [TS05]. Furthermore, several recent papers have proposed optimization algorithms for linear models using ℓ_1 [CKL$^+$10, SH10] and ℓ_2 [VB10] fusion penalties. This paper extends this line of work by developing a family of fusion penalties that results in the "clusterpath," a hierarchical regularization path which is useful for clustering problems.

1.1 Motivation by relaxing hierarchical clustering

Hierarchical or agglomerative clustering is calculated using a greedy algorithm, which for n points in $\mathbb{R}^p$ recursively joins the points which are closest together until all points are joined. For the data matrix $X \in \mathbb{R}^{n\times p}$ this suggests the optimization problem

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^{n\times p}} \quad & \frac{1}{2}||\alpha - X||_F^2 \\ \text{subject to} \quad & \sum_{i<j} 1_{\alpha_i\neq\alpha_j} \leq t, \end{aligned} \tag{1}$$

where $||\cdot||_F^2$ is the squared Frobenius norm, $\alpha_i \in \mathbb{R}^p$ is row i of α, and $1_{\alpha_i\neq\alpha_j}$ is 1 if $\alpha_i \neq \alpha_j$, and 0 otherwise. We use the notation $\sum_{i<j} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}$ to sum over all the $n(n-1)/2$ pairs of data points. Note that when we fix $t \geq n(n-1)/2$ the problem is unconstrained and the solutions are $\alpha_i = X_i$ for all i. If $t = n(n-1)/2-1$, we force one pair of coefficients to fuse, and this is equivalent to the first step in hierarchical clustering. Furthermore, when $t = 0$, the solutions are clearly $\alpha_i = \bar{X} = \sum_{i=1}^{n} X_i/n$. In general this is a difficult combinatorial optimization problem.

Instead, we propose a convex relaxation, which results in the family of optimization problems defined by

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^{n\times p}} \quad & \frac{1}{2}||\alpha - X||_F^2 \\ \text{subject to} \quad & \Omega_q(\alpha) = \sum_{i<j} w_{ij}||\alpha_i - \alpha_j||_q \leq t, \end{aligned} \tag{2}$$

where $w_{ij} > 0$, and $||\cdot||_q$, $q \in \{1, 2, \infty\}$ is the ℓ_q-norm on $\mathbb{R}^p$, which will induce sparsity in the differences of the rows of α. When rows fuse we say they form a cluster, and the continuous regularization path of optimal solutions formed by varying t is what we call the "clusterpath."

This parameterization in terms of t is cumbersome when comparing datasets since we take $0 \leq t \leq \Omega_q(X)$, so we introduce the following parametrization with $0 \leq s \leq 1$:

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^{n\times p}} \quad & \frac{1}{2}||\alpha - X||_F^2 \\ \text{subject to} \quad & \Omega_q(\alpha)/\Omega_q(X) \leq s. \end{aligned} \tag{3}$$

The equivalent Langrangian dual formulation will also be convenient for optimization algorithms:

$$\min_{\alpha\in\mathbb{R}^{n\times p}} f_q(\alpha, X) = \frac{1}{2}||\alpha - X||_F^2 + \lambda\Omega_q(\alpha). \tag{4}$$

Equivalent parameter values and their solutions for the ends of the path are compared in Table 1.

Table 1: Parameters and solutions for the ends of the clusterpath. Note that in general, there is no closed form expression for λ_{max}, and $\bar{X} = \sum_{i=1}^{n} X_i/n$.

t	s	λ	α_i^*
$\Omega(X)$	1	0	X_i
0	0	λ_{max}	$\bar{X}$

The above optimization problems require the choice of predefined, pair-specific weights $w_{ij} > 0$, which can be used to control the geometry of the solution path. In most of our experiments we use weights that decay with the distance between points $w_{ij} = \exp(-\gamma||X_i - X_j||_2^2)$, which results in a clusterpath that is sensitive to local density in the data. Another choice for the weights is $w_{ij} = 1$, which allows efficient computation of the ℓ_1 clusterpath (§2.2). Choosing weights based on some supplemenary data space $w_{ij} = \exp(-\gamma||Y_i - Y_j||^2)$ could also be interesting. For example, for clustering pixels of an image into objects, we could take $X \in \mathbb{R}^{n\times 2}$ to be the matrix of pixel positions, and $Y \in \mathbb{R}^{n\times p}$ to be the matrix of visual features for each pixel.

1.2 Visualizing the geometry of the clusterpath

This optimization problem has an equivalent geometric interpretation (Figure 1). For the identity weights $w_{ij} = 1$, the solution corresponds to the closest points α to the points X, subject to a constraint on the sum of distances between pairs of points. For general weights, we constrain the total area of the rectangles of width w_{ij} between pairs of points.

In this work we develop dedicated algorithms for solving the clusterpath which allow scaling to large data, but initially we used `cvxmod` for small problems [MB08], as the authors do in a similar independent formulation [LOL11].

We used `cvxmod` to compare the geometry of the clusterpath for several choices of norms and weights (Figure 2). Note the piecewise linearity of the ℓ_1 and ℓ_∞ clusterpath, which can be exploited to find the solutions using efficient path-following homotopy algorithms. Furthermore, it is evident that the ℓ_2 path is invariant to rotation of the input data X, whereas the others are not.

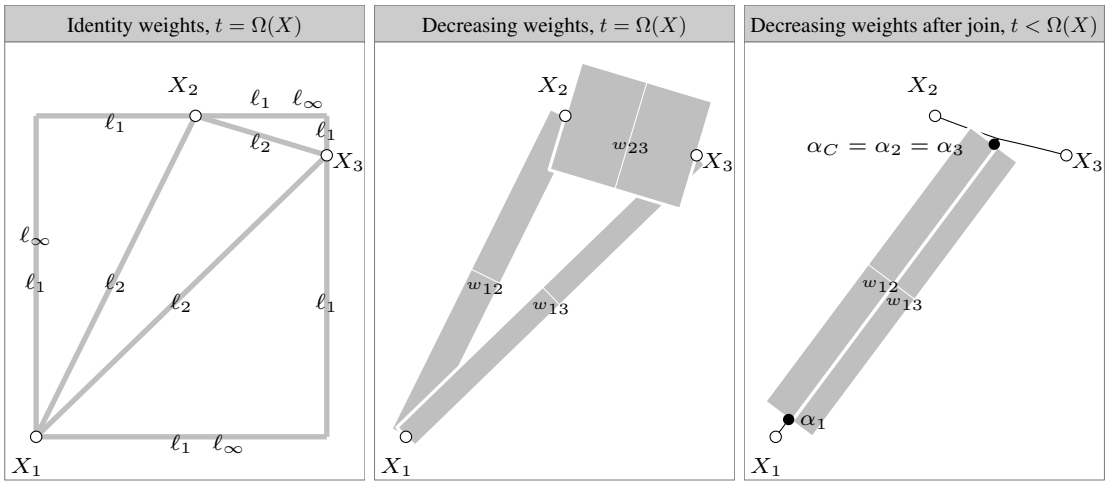

Figure 1: Geometric interpretation of the optimization problem (2) for data $X \in \mathbb{R}^{3\times 2}$. **Left:** with the identity weights $w_{ij} = 1$, the constraint $\Omega_q(\alpha) = \sum_{i<j} w_{ij}||\alpha_i - \alpha_j||_q \leq t$ is the ℓ_q distance between all pairs of points, shown as grey lines. **Middle:** with general weights w_{ij}, the ℓ_2 constraint is the total area of rectangles between pairs of points. **Right:** after constraining the solution, α_2 and α_3 fuse to form the cluster C, and the weights are additive: $w_{1C} = w_{12} + w_{13}$.

2 Optimization

2.1 A homotopy algorithm for the ℓ_1 solutions

For the problem involving the ℓ_1 penalty, we first note that the problem is separable on dimensions. The cost function can be written as

$$\begin{aligned} f_1(\alpha, X) &= \frac{1}{2}||\alpha - X||_F^2 + \lambda\Omega_1(\alpha) \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{p}(\alpha_{ik} - X_{ik})^2 + \lambda\sum_{i<j} w_{ij}\sum_{k=1}^{p}|\alpha_{ik} - \alpha_{jk}| \\ &= \sum_{k=1}^{p}\left[\frac{1}{2}\sum_{i=1}^{n}(\alpha_{ik} - X_{ik})^2 + \lambda\sum_{i<j} w_{ij}|\alpha_{ik} - \alpha_{jk}|\right] \\ &= \sum_{k=1}^{p} f_1(\alpha^k, X^k), \end{aligned}$$

where $\alpha^k \in \mathbb{R}^n$ is the k-th column from α. Thus, solving the minimization with respect to the entire matrix X just amounts to solving p separate minimization subproblems:

$$\min_{\alpha\in\mathbb{R}^{n\times p}} f_1(\alpha, X) = \sum_{k=1}^{p}\min_{\alpha^k\in\mathbb{R}^n} f_1(\alpha^k, X^k).$$

For each of these subproblems, we can exploit the FLSA path algorithm [Hoe09]. This is a homotopy algorithm similar to the LARS that exploits the piecewise linearity of the path to very quickly calculate the entire set of solutions [EHJT04].

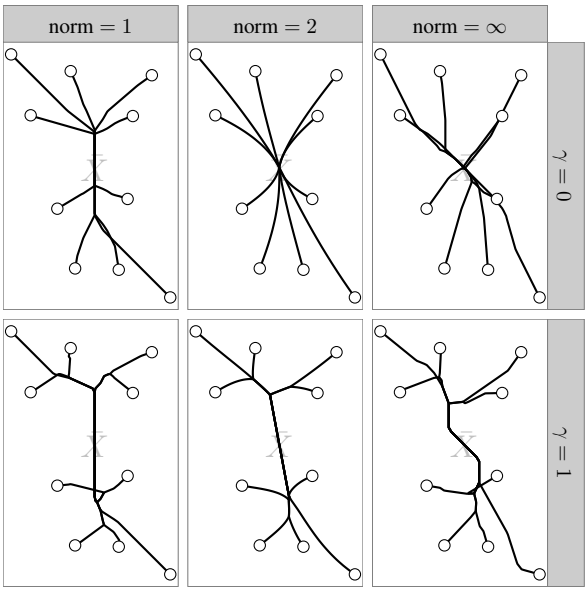

Figure 2: Some random normal data $X \in \mathbb{R}^{10\times 2}$ were generated (white dots) and their mean $\bar{X}$ is marked in the center. The clusterpath (black lines) was solved using `cvxmod` for 3 norms (panels from left to right) and 2 weights (panels from top to bottom), which were calculated using $w_{ij} = \exp(-\gamma||X_i - X_j||^2)$. For $\gamma = 0$, we have $w_{ij} = 1$.

In the LARS, variables jump in and out the active set, and we must check for these events at each step in the path. The analog in the FLSA path algorithm is the necessity to check for cluster splits, which occur when the optimal solution path requires unfusing a pair coefficients. Cluster splits were not often observed on our experiments, but are also possible for the ℓ_2 clusterpath, as illustrated in Figure 4. The FLSA path algorithm checks for a split of a cluster of size n_C by solving a max-flow problem using a push-relabel algorithm, which has complexity $O(n_C^3)$ [CLRS01]. For large data sets, this can be prohibitive, and for any clustering algorithm, splits make little sense.

One way around this bottleneck is to choose weights w in a way such that no cluster splits are possible in the path. The modified algorithm then only considers cluster joins, and results in a complexity of $O(n \log n)$ for a single dimension, or $O(pn \log n)$ for p dimensions. One choice of weights that results in no cluster splits is the identity weights $w_{ij} = 1$, which we prove below.

2.2 The ℓ_1 clusterpath using $w_{ij} = 1$ contains no splits

The proof will establish a contradiction by examining the necessary conditions on the optimal solutions during a cluster split. We will need the following lemma.

Lemma 1. *Let $C = \{i : \alpha_i = \alpha_C\} \subseteq \{1, ..., n\}$ be the cluster formed after the fusion of all points in C. At any point in the regularization path, the slope of its coefficient is given by*

$$v_C = \frac{d\alpha_C}{d\lambda} = \frac{1}{|C|} \sum_{j \notin C} w_{Cj} \operatorname{sign}(\alpha_j - \alpha_C). \tag{5}$$

Proof. Consider the following sufficient optimality condition, for all $i = 1, \ldots, n$:

$$0 = \alpha_i - X_i + \lambda \sum_{\substack{j \neq i \\ \alpha_i \neq \alpha_j}} w_{ij} \operatorname{sign}(\alpha_i - \alpha_j) + \lambda \sum_{\substack{j \neq i \\ \alpha_i = \alpha_j}} w_{ij} \beta_{ij},$$

with $|\beta_{ij}| \leq 1$ and $\beta_{ij} = -\beta_{ji}$ [Hoe09]. We can rewrite the optimality condition for all $i \in C$:

$$0 = \alpha_C - X_i + \lambda \sum_{j \notin C} w_{ij} \operatorname{sign}(\alpha_C - \alpha_j) + \lambda \sum_{i \neq j \in C} w_{ij} \beta_{ij}.$$

Furthermore, by summing each of these equations, we obtain the following:

$$\alpha_C = \bar{X}_C + \frac{\lambda}{|C|} \sum_{j \notin C} w_{Cj} \operatorname{sign}(\alpha_j - \alpha_C),$$

where $\bar{X}_C = \sum_{i \in C} X_i / |C|$ and $w_{Cj} = \sum_{i \in C} w_{ij}$. Taking the derivative with respect to λ gives us the slope v_C of the coefficient line for cluster C, proving Lemma 1. □

We will use Lemma 1 to prove by contradiction that cluster splitting is impossible for the case $w_{ij} = 1$ for all i and j.

Theorem 1. *Taking $w_{ij} = 1$ for all i and j is sufficient to ensure that the ℓ_1 clusterpath contains no splits.*

Proof. Consider at some λ the optimal solution α, and let C be a cluster of any size among these optimal solutions. Denote the set $\overline{C} = \{i : \alpha_i > \alpha_C\}$ the set of indices of all larger optimal coefficients and $\underline{C} = \{i : \alpha_i < \alpha_C\}$ the set of indices of all smaller optimal coefficients. Note that $\overline{C} \cup \underline{C} \cup C = \{1, \ldots, n\}$.

Now, assume C splits into C_1 and C_2 such that $\alpha_1 > \alpha_2$. By Lemma 1, if this situation constitutes an optimal solution, then the slopes are:

$$\begin{aligned} v_{C_1} &= \frac{1}{|C_1|} \left(\sum_{j \in \overline{C}} w_{jC_1} - \sum_{j \in C_2} w_{jC_1} - \sum_{j \in \underline{C}} w_{jC_1} \right) \\ v_{C_2} &= \frac{1}{|C_2|} \left(\sum_{j \in \overline{C}} w_{jC_2} + \sum_{j \in C_1} w_{jC_2} - \sum_{j \in \underline{C}} w_{jC_2} \right). \end{aligned}$$

For the identity weights, this simplifies to

$$\begin{aligned} v_{C_1} &= |\overline{C}| - |C_2| - |\underline{C}| \\ v_{C_2} &= |\overline{C}| + |C_1| - |\underline{C}|. \end{aligned}$$

Thus $v_{C_1} < v_{C_2}$ which contradicts the assumption that $\alpha_1 > \alpha_2$, forcing us to conclude that no split is possible for the identity weights. □

Thus the simple FLSA algorithm of complexity $O(n \log n)$ without split checks is sufficient to calculate the ℓ_1 clusterpath for the identity weights, as shown in Figure 3.

Furthermore, since the clusterpath is strictly agglomerative on each dimension, it is also strictly agglomerative when independently applied to each column of a matrix of data. Thus the ℓ_1 clusterpath for a matrix of data is strictly agglomerative, and results in an algorithm of complexity $O(pn \log n)$. This is an interesting alternative to hierarchical clustering, which normally requires $O(pn^2)$ space and time for $p > 1$. The ℓ_1 clusterpath can be used when n is very large, and hierarchical clustering is not feasible.

The proposed homotopy algorithm only gives solutions to the ℓ_1 clusterpath for identity weights, but since the ℓ_1 clusterpath in 1 dimension is a special case of the ℓ_2 clusterpath, the algorithms proposed in the next subsection also apply to solving the ℓ_1 clusterpath with general weights.

2.3 An active-set descent algorithm for the ℓ_2 solutions

For the ℓ_2 problem, we have the following cost function:

$$f_2(\alpha, X) = \frac{1}{2}||\alpha - X||_F^2 + \lambda \Omega_2(\alpha),$$

A subgradient condition sufficient for an optimal α is for all $i \in 1, ..., n$:

$$0 = \alpha_i - X_i + \lambda \sum_{\substack{j \neq i \\ \alpha_j \neq \alpha_i}} w_{ij} \frac{\alpha_i - \alpha_j}{||\alpha_i - \alpha_j||_2} + \lambda \sum_{\substack{j \neq i \\ \alpha_j = \alpha_i}} w_{ij} \beta_{ij},$$

with $\beta_{ij} \in \mathbb{R}^p$, $||\beta_{ij}||_2 \leq 1$ and $\beta_{ij} = -\beta_{ji}$. Summing over all $i \in C$ gives the subgradient for the cluster C:

$$G_C = \alpha_C - \bar{X}_C + \frac{\lambda}{|C|} \sum_{j \notin C} w_{Cj} \frac{\alpha_C - \alpha_j}{||\alpha_C - \alpha_j||_2}, \tag{6}$$

where $\bar{X}_C = \sum_{i \in C} X_i / |C|$ and $w_{Cj} = \sum_{i \in C} w_{ij}$.

To solve the ℓ_2 clusterpath, we propose a subgradient descent algorithm, with modifications to detect cluster fusion and splitting events (Algorithm 1). Note that due to the continuity of the ℓ_2 clusterpath, it is advantageous to use warm restarts between successive calls to SOLVE-L2, which we do using the values of α and $clusters$.

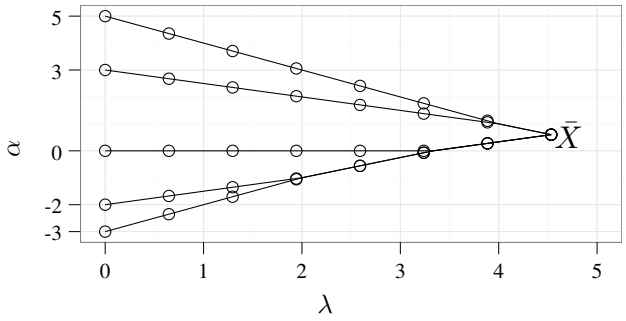

Figure 3: The ℓ_1 clusterpath calculated using the homotopy algorithm (lines) and cvxmod (points) for an $X \in \mathbb{R}^5$ and $w_{ij} = 1$.

Algorithm 1 CLUSTERPATH-L2

Input: data $X \in \mathbb{R}^{n \times p}$, weights $w_{ij} > 0$, starting $\lambda > 0$
$\alpha \leftarrow X$
$clusters \leftarrow \{\{1\}, ..., \{n\}\}$
while $|\ clusters\ | > 1$ **do**
  $\alpha, clusters \leftarrow$ SOLVE-L2$(\alpha, clusters, X, w, \lambda)$
  $\lambda \leftarrow \lambda \times 1.5$
  if we are considering cluster splits **then**
    $clusters \leftarrow \{\{1\}, ..., \{n\}\}$
  end if
end while
return table of all optimal α and λ values.

Surprisingly, the ℓ_2 path is not always agglomerative, and in this case to reach the optimal solution requires restarting $clusters = \{\{1\}, ..., \{n\}\}$. The clusters will rejoin in the next call to SOLVE-L2 if necessary. This takes more time but ensures that the optimal solution is found, even if there are splits in the clusterpath, as in Figure 4.

We conjecture that there exist certain choices of w for which there are no splits in the ℓ_2 clusterpath. However, a theorem analogous to Theorem 1 that establishes necessary and sufficient conditions on w and X for splits in the ℓ_2 clusterpath is beyond the scope of this article. We have not observed cluster splits in our calculations of the path for identity weights $w_{ij} = 1$ and decreasing weights $w_{ij} = \exp(-\gamma||X_i - X_j||_2^2)$, and we conjecture that these weights are sufficient to ensure no splits.

SUBGRADIENT-L2 calculates the subgradient from (6), for every cluster $C \in clusters$.

We developed 2 approaches to implement SUBGRADIENT-STEP. In both cases we use the update $\alpha \leftarrow \alpha - rG$. With decreasing step size $r = 1/\ iteration$, the algorithm takes many steps before converging to the optimal solution, even though we restart the iteration count after cluster fusions. The second approach we used is a line search. We evaluated the cost function at several points r and picked the r with the lowest cost. In practice, we observed fastest performance when we alternated every other step between decreasing and line search.

DETECT-CLUSTER-FUSION calculates pairwise differences between points and checks for cluster fusions, returning the updated matrix of points α and the new list of clusters. When 2 clusters C_1 and C_2 fuse to produce

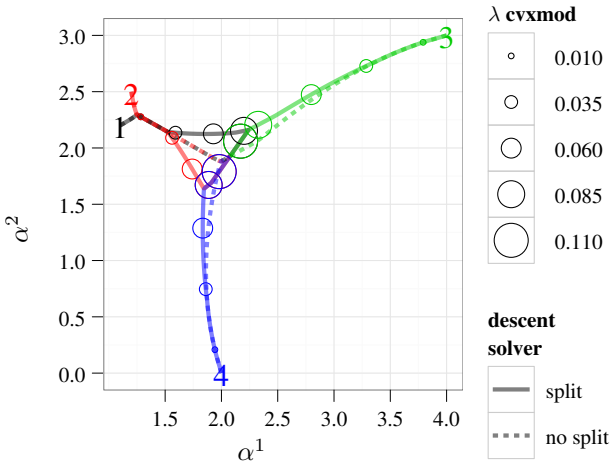

Figure 4: An example of a split in the ℓ_2 clusterpath for $X \in \mathbb{R}^{4 \times 2}$. Data points are labeled with numbers, the CLUSTERPATH-L2 is shown as lines, and solutions from `cvxmod` are shown as circles. $w_{12} = 9$, $w_{13} = w_{24} = 20$, and $w_{ij} = 1$ for the others (best seen in color).

Algorithm 2 SOLVE-L2

Input: initial guess α, initial $clusters$, data X, weights w, regularization λ
$G \leftarrow$ SUBGRADIENT-L2$(\cdot)$
while $||G||_F^2 > \epsilon_{\text{opt}}$ **do**
  $\alpha \leftarrow$ SUBGRADIENT-STEP$(\cdot)$
  $\alpha, clusters \leftarrow$ DETECT-CLUSTER-FUSION$(\cdot)$
  $G \leftarrow$ SUBGRADIENT-L2$(\cdot)$
end while
return $\alpha, clusters$

a new cluster C, the coefficient of the new cluster is calculated using the weighted mean:

$$\alpha_C = \frac{|C_1|\alpha_{C_1} + |C_2|\alpha_{C_2}}{|C_1| + |C_2|}. \tag{7}$$

We developed 2 methods to detect cluster fusions. First, we can simply use a small threshhold on $||\alpha_{C_1} - \alpha_{C_2}||_2$, which we usually take to be some fraction of the smallest nonzero difference in the original points $||X_i - X_j||_2$. Second, to confirm that the algorithm does not fuse points too soon, for each possible fusion, we checked if the cost function decreases. This is similar to the approach used by [FHHT07], who use a coordinate descent algorithm to optimize a cost function with an ℓ_1 fusion penalty. Although this method ensures that we reach the correct solution, it is quite slow since it requires evaluation of the cost function for every possible fusion event.

2.4 The Frank-Wolfe algorithm for ℓ_∞ solutions

We consider the following ℓ_∞ problem:

$$\min_{\alpha \in \mathbb{R}^{n \times p}} f_\infty(\alpha, X) = \frac{1}{2}||\alpha - X||_F^2 + \lambda \Omega_\infty(\alpha). \tag{8}$$

This problem has a piecewise linear regularization path which we can solve using a homotopy algorithm to exactly calculate all the breakpoints [RZ07, ZRY09]. However, empirically, the number of breakpoints in the path grows fast with p and n, leading to instability in the homotopy algorithm.

Instead, we show show that our problem is equivalent to a norm minimization over a polytope, for which an efficient algorithm exists [FW56].

Using the dual formulation of the ℓ_∞ norm, the regularization term is equal to:

$$\Omega_\infty(\alpha) = \sum_{i<j} w_{ij} \max_{\substack{s_{ij} \in \mathbb{R}^p \\ ||s_{ij}||_1 \leq 1}} s_{ij}^T(\alpha_i - \alpha_j).$$

Denoting by $r_i = \sum_{j>i} s_{ij} w_{ij} - \sum_{j<i} s_{ji} w_{ij} \in \mathbb{R}^p$, and by $\mathcal{R}$ the set of constraints over $R = (r_1, \ldots, r_n)$ such that the constraints over s_{ij} are respected, we have:

$$\Omega_\infty(\alpha) = \max_{R \in \mathcal{R}} \operatorname{tr}\left(R^T \alpha\right).$$

Since $\mathcal{R}$ is defined as a set of linear combinations of ℓ_1-ball inequalities, $\mathcal{R}$ is a polytope. Denoting by $Z = X - \lambda R$ and $\mathcal{Z} = \{Z \mid \frac{1}{\lambda}(X - Z) \in \mathcal{R}\}$, it is straightforward to prove that problem (8) is equivalent to:

$$\min_{\alpha \in \mathbb{R}^{n \times p}} \max_{Z \in \mathcal{Z}} H(\alpha, Z) = \|\alpha - Z\|_F^2 - \|Z\|_F^2,$$

where strong duality holds [BV03]. For a given Z, the minimum of H in α is obtained by $\alpha = Z$, leading to a norm minimization over the polytope $\mathcal{Z}$.

$$\min_{Z \in \mathcal{Z}} \frac{1}{2}\|Z\|_F^2. \tag{9}$$

This problem can be solved efficiently by using the Frank-Wolfe algorithm [FW56]. This algorithm to minimize a quadratic function over a polytope may be used as soon as it is possible to minimize linear functions in closed form. It is also known as the minimum-norm-point algorithm when applied to submodular function minimization [FHI06]. In practice, it is several orders of magnitude faster than other common discrete optimization algorithms, but there is no theoretical guarantee on its complexity [KG09].

3 The spectral clusterpath: a completely convex formulation of spectral clustering

For spectral clustering, the usual formulation uses eigenvectors of the normalized Laplacian as the inputs to a standard clustering algorithm like k-means [NJW01]. Specifically, for several values of γ, we compute a pairwise affinity matrix W such that $W_{ij} = \exp(-\gamma||X_i - X_j||_2^2)$ and a Laplacian matrix $L = D - W$ where D is a diagonal matrix such that $D_{ii} = \sum_{j=1}^{n} W_{ij}$. For each value of γ, we run k-means on the normalized eigenvectors associated with k smallest eigenvalues of L, then keep the γ with lowest reconstruction error.

Some instability in spectral clustering may come from the following 2 steps. First, the matrix of eigenvectors is formed by hard-thresholding the eigenvalues, which is unstable when several eigenvalues are close. Second, the clusters are located using the k-means algorithm, which attempts to minimize a non-convex objective. To relax these potential sources of instability, we propose the "spectral clusterpath," which replaces (a) hard-thresholding by soft-thresholding and (b) k-means by the clusterpath.

Concretely, we call $(\Lambda_i)_{1 \leq i \leq n}$ the nontrivial eigenvalues sorted in ascending order, and we write the matrix of transformed eigenvectors to cluster as VE, where V is the full matrix of sorted nontrivial eigenvectors and E is the diagonal matrix such that $E_{ii} = e(\Lambda_i)$, and $e : \mathbb{R} \to \mathbb{R}$ ranks importance of eigenvectors based on their eigenvalues. Standard spectral clustering takes $e_{01}(x) = 1_{x \leq \Lambda_k}$ such that only the first k eigenvalues are selected. This is a non-convex hard-thresholding of the full matrix of eigenvectors. We propose the exponential function $e_{\exp}(x) = \exp(-\nu x)$, with $\nu > 0$, as a convex relaxation. This smooth decreasing function continuously scales the eigenvectors based on the eigenvalues.

4 Results

Our model poses 3 free parameters to choose for each matrix to cluster: norm, weights, and regularization. On one hand, this offers the flexibility to tailor the geometry of the solution path and number of clusters for each data set. On the other hand, this poses model selection problems as training clustering models is not straightforward. Many heuristics have been proposed for automatically choosing the number of clusters [TWH01], but it is not clear which of these is applicable to any given data set.

In the experiments that follow, we chose the model based on the desired geometry of the solution path and number of clusters. We generally expect rotation invariance in multivariate clustering models, so we chose the ℓ_2 norm with Gaussian weights to encourage sensitivity to local density.

Table 2: Mean and standard deviation of performance and timing of several clustering methods on identifying 20 simulations of the half-moons in Figure 5. Ng et al. uses $\tilde{L} = I - D^{-1/2} W D^{-1/2}$ rather than $L = D - W$ as discussed in the text. Large Rand means good clustering.

Clustering method	Rand	SD	Seconds	SD
$e_{\exp}$ spectral clusterpath	0.9967	0.0073	8.4920	2.6474
$e_{\exp}$ spectral kmeans	0.9967	0.0073	3.1078	0.0848
clusterpath	0.9580	0.1262	29.4738	2.3122
e_{01} Ng et al. kmeans	0.9538	0.1911	7.3772	0.4222
e_{01} spectral kmeans	0.9149	0.1978	3.2647	0.2129
Gaussian mixture	0.4254	0.1334	0.0783	0.0051
average linkage	0.4047	0.1341	0.0557	0.0009
kmeans	0.2668	0.0466	0.0011	0.0002

4.1 Verification on non-convex half-moon clusters

To compare our algorithm to other popular methods in the setting of non-convex clusters, we generated data in the form of 2 interlocking half-moons (Figure 5), which we used as input for several clustering algorithms (Table 2). We used the original data as input for k-means, Gaussian mixtures, average linkage hierarchical clustering, and the ℓ_2 clusterpath with $\gamma = 2$. For the other methods, we use the eigenvectors from spectral clustering as input. Each algorithm uses 2 clusters and performance is measured using the normalized Rand index, which varies from 1 for a perfect match to 0 for completely random assignment [HA85].

In the original input space, hierarchical clustering and k-means fail, but the clusterpath is able to identify the clusters as well as the spectral methods, and has the added benefit of learning a tree from the data. However, the clusterpath takes 3-10 times more time than the spectral methods. Of the methods that cluster the eigenvectors, the most accurate 2 methods use $e_{\exp}$ rather than e_{01}, providing evidence that the convex relaxation stabilizes the clustering.

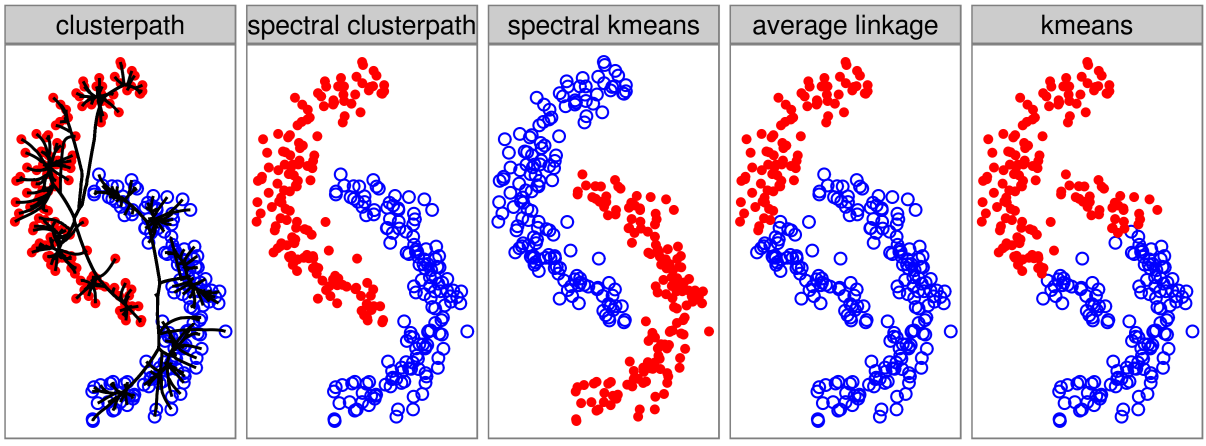

Figure 5: Typical results for 5 clustering algorithms applied to 2 half-moon non-convex clusters. The ℓ_2 clusterpath tree learned from the data is also shown. Spectral clustering and the clusterpath correctly identify the clusters, while average linkage hierarchical clustering and k-means fail.

Table 3: Performance of several clustering methods on identifying a grid of Gaussian clusters. Means and standard deviations from 20 simulations are shown.

Clustering method	Rand	SD
kmeans	0.8365	0.0477
clusterpath	0.9955	0.0135
average linkage hierarchical	1.0000	0.0000

4.2 Recovery of many Gaussian clusters

We also tested our algorithm in the context of 25 Gaussian clusters arranged in a 5×5 grid in 2 dimensions (Figure 6). 20 data points were generated from each cluster, and the resulting data were clustered using k-means, hierarchical clustering, and the weighted ℓ_2 clusterpath. The clusterpath performs similarly to hierarchical clustering, which exactly recovers the clusters, and k-means fails. Thus, the clusterpath may be useful for clustering tasks that involve many clusters.

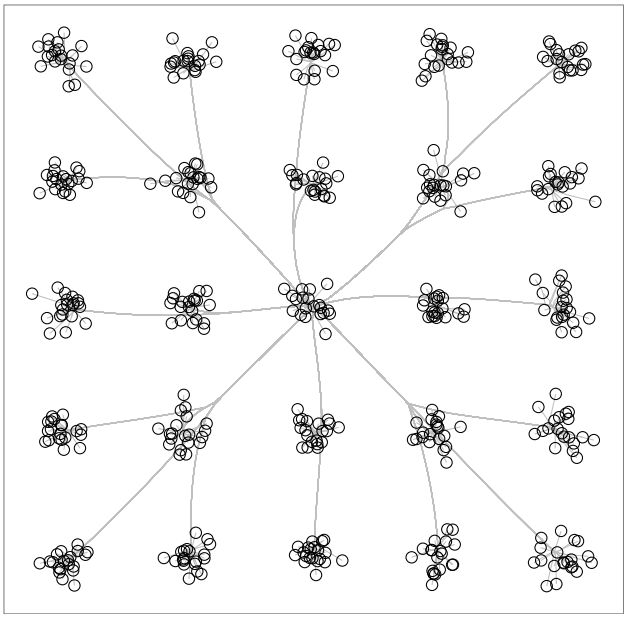

Figure 6: A grid of gaussian clusters breaks k-means but not the clusterpath. The weighted ℓ_2 clusterpath is shown as grey lines.

4.3 Application to clustering the iris data

To evaluate the clusterpath on a nontrivial task, we applied it and other common clustering methods to the scaled iris data (Figure 7). We calculated a series of clusterings using each algorithm and measured performance of each using the normalized Rand index (Figure 8).

The iris data have 3 classes, of which 2 overlap, so the Gaussian Mixture Model is the only algorithm capable of accurately detecting these clusters when $k = 3$. These data suggest that the clusterpath is not suitable for detecting clusters with large overlap. However, performance is as good as hierarchical clustering, less variable than k-means, and more stable as the number of clusters increases.

Additionally, Figure 8 shows that the clusterpath classification accuracy on the moons data increases as we increase the weight parameter γ.

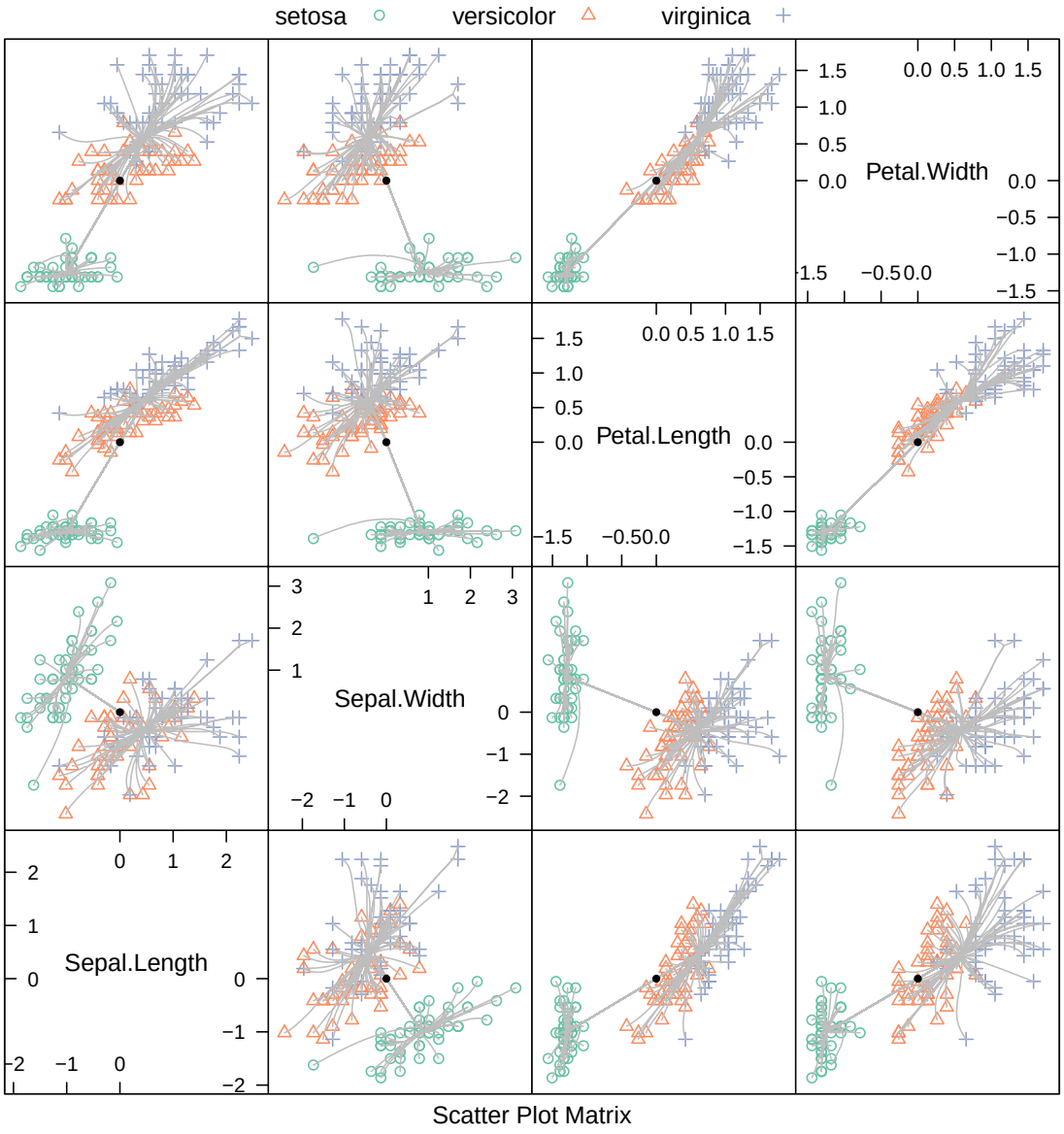

Figure 7: The 4-dimensional iris data were scaled, and the ℓ_2 clusterpath was calculated using Gaussian weights with $\gamma = 1$, and plotted using grey lines. The mean of the data is shown as a black dot.

5 Conclusions

We proposed a framework for clustering using linear models and several convex pairwise fusion penalties, which lead to efficient algorithms for optimization (Table 4). The ℓ_1 path-following homotopy algorithm easily scales to thousands of points. The other proposed algorithms can be directly applied to hundreds of points, and could be applied to larger datasets by, for example, adding a preprocessing step using k-means. The algorithms were implemented in R, C++, and MATLAB, and will be published soon.

Our experiments demonstrated the flexibility of the ℓ_2 clusterpath for the unsupervised learning of non-convex clusters, large numbers of clusters, and hierarchical structures. We also observed that relaxing hard-thresholding in spectral clustering is useful for increasing clustering accuracy and stability. For the iris data, the clusterpath performed as well as hierarchical clustering, and is more stable than k-means.

We proved that the identity weights are sufficient for the ℓ_1 clusterpath to be strictly agglomerative. Establishing necessary and sufficient conditions on the weights for the ℓ_2 problem is an avenue for further research.

To extend these results, we are currently pursuing research into optimizing a linear model with a non-identity design matrix and the clusterpath penalty. We note that there could be a future application for the algorithms presented in this article in solving the proximal operator, which is the same as (4) for the clusterpath penalty.

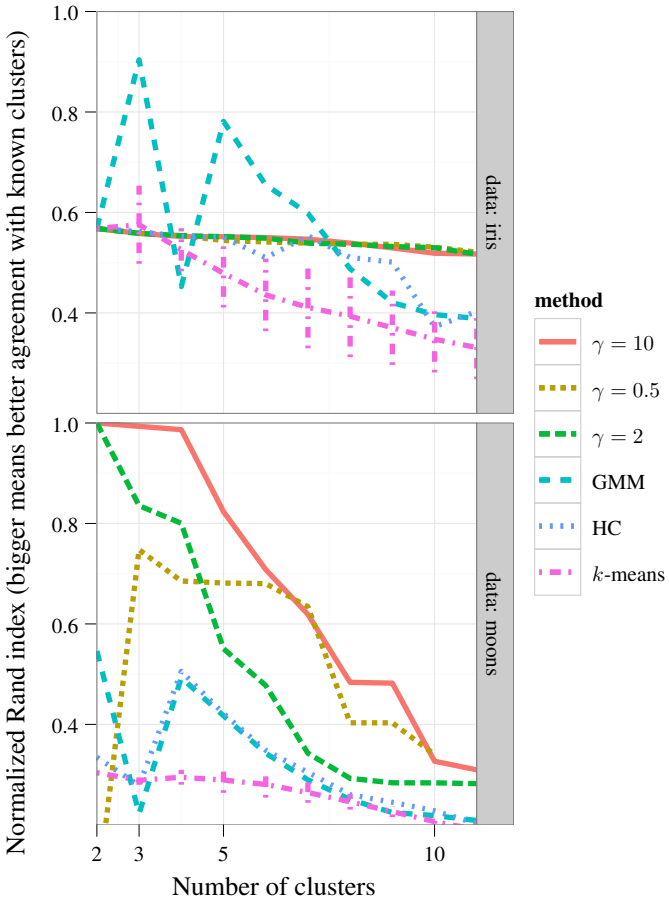

Figure 8: Performance on the iris and moons data, as measured by the normalized Rand index of models with 2-11 clusters. The weighted ℓ_2 clusterpath was calculated using 3 different Gaussian weight parameters γ, and we compare with Gaussian Mixture Models (GMM), Hierarchical Clustering (HC), and k-means.

Table 4: Algorithms proposed to solve the clusterpath.

Norm	Properties	Algorithm	Complexity	Problem sizes
1	piecewise linear, separable	homotopy	$O(pn \log n)$	large $\approx 10^5$
2	rotation invariant	active-set descent	$O(n^2 p)$	medium $\approx 10^3$
∞	piecewise linear	Frank-Wolfe	unknown	medium $\approx 10^3$

Awknowledgements

FB was supported by grant SIERRA-ERC-239993. TDH was supported by grant DIGITEO-BIOVIZ-2009-25D. JPV was supported by grants ANR-07-BLAN-0311-03 and ANR-09-BLAN-0051-04.

References

[BH08] F. Bach and Zaïd Harchoui. DIFFRAC: a discriminative and flexible framework for clustering. In *Adv. NIPS*, 2008.

[BV03] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge U. P., 2003.

[CKL+10] X. Chen, S. Kim, Q. Lin, J. G. Carbonell, and E. P. Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso, 2010. arXiv:1005.3579.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.

[EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of statistics*, 32(2):40–99, 2004.

[FHHT07] J. Friedman, T. Hastie, H. Hoefling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):30–32, 2007.

[FHI06] S. Fujishige, T. Hayashi, and S. Isotani. The minimum-norm-point algorithm applied to submodular function minimization and linear programming, 2006. RIMS No 1571. Kyoto University.

[FW56] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95110, 1956.

[HA85] L. Hubert and P. Arabie. Comparing partitions. *J. Classification*, 2:193–218, 1985.

[Hoe09] H. Hoefling. A path algorithm for the fused lasso signal approximator. arXiv:0910.0526, 2009.

[KG09] A. Krause and C. Guestrin. Beyond convexity: Submodularity in machine learning. In *IJCAI*, 2009.

[LOL11] Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. Clustering using sum-of-norms regularization; with application to particle filter output computation. Technical Report LiTH-ISY-R-2993, Department of Electrical Engineering, Linkping University, February 2011.

[MB08] J. Mattingley and S. Boyd. CVXMOD: Convex optimization software in Python (web page and software), July 2008.

[NJW01] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Adv. NIPS*, 2001.

[RZ07] S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *Annals of Statistics*, 35(3):1012–1030, 2007.

[SH10] X. Shen and H.-C. Huang. Grouping pursuit through a regularization solution surface. *Journal of the American Statistical Association*, 105(490):727–739, 2010.

[Tib96] R. Tibshirani. Regression Shrinkage and Selection Via the Lasso. *J. R. Statist. Soc. B.*, 58(1):267–288, 1996.

[TS05] R. Tibshirani and M. Saunders. Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc. B.*, 67:9–08, 2005.

[TWH01] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *J. R. Statist. Soc. B*, 63:41–23, 2001.

[VB10] J.-P. Vert and K. Bleakley. Fast detection of multiple change-points shared by many signals using group LARS. In *Adv. NIPS*, 2010.

[XNLS04] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *Adv. NIPS*, 2004.

[YL06] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, 68(B):4–7, 2006.

[ZRY09] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *Ann. Stat.*, 37(6A):3468–3497, 2009.